INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

An Environmental Feature Representation for Robust Speech Recognition and for Environment Identification

Xue Feng[1], *Brigitte Richardson*[2], *Scott Amman*[2], *James Glass*[1]

[1]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, USA, 02139
[2]Ford Motor Company

xfeng@csail.mit.edu, {bricha46, samman}@ford.com, glass@mit.edu

Abstract

In this paper we investigate environment feature representations, which we refer to as e-vectors, that can be used for environment adaption in automatic speech recognition (ASR), and for environment identification. Inspired by the fact that i-vectors in the total variability space capture both speaker and channel environment variability, our proposed e-vectors are extracted from i-vectors. Two extraction methods are proposed: one is via linear discriminant analysis (LDA) projection, and the other via a bottleneck deep neural network (BN-DNN). Our evaluations show that by augmenting DNN-HMM ASR systems with the proposed e-vectors for environment adaptation, ASR performance is significantly improved. We also demonstrate that the proposed e-vector yields promising results on environment identification.

Index Terms: robust speech recognition, environmental features, noise adaptation, speech environment identification

1. Introduction

There is a continuously growing demand for hands-free speech input for various applications [1, 2]. However, in the above distant-talking speech communication systems, the presence of environmental noise, and/or reverberation, often causes a dramatic performance drop on automatic speech recognition (ASR) systems. When speech signals are corrupted by noise, the statistics of speech features extracted from the speech signals are also distorted in various domains, and the ASR performance is significantly degraded. Therefore, there is a strong need for noise-aware features that can improve robustness in noisy speech recognition system.

I-vector has been extracted by factor analysis and successfully used for speaker recognition and speaker adaptation [3, 4, 5], and demonstrated state-of-the-art performance for text-independent speaker detection tasks in the NIST speaker recognition evaluations.

In contrast to i-vector's success and popularity in speaker related tasks, there has been little research on its usefulness in channel and environment applications. I-vectors are extracted in a way that makes no distinction between channel and speaker variability. Inspired by this fact, we propose features derived from i-vector in the total variability space to capture environmental variability only.

One way of separating the environment related information from i-vector is via dimensionality reduction, using method including linear discriminant analysis (LDA). We refer to the environmental feature obtained by this method as lda-evector.

We also propose a second method to extract environmental feature from i-vector, by training a bottleneck neural network (BN-NN). In 2011, Yu and Seltzer applied a DNN for extracting BN features, with the bottleneck being a small hidden layer placed in the middle of the network [6]. Bottleneck features have shown success in speaker adaptation and language identification, as in [7, 8].

Having the novel feature representation extracted from total variability space, we then perform environment adaptation using this new feature, in conjunction with acoustic features, for speech recognition. Our results demonstrate that for a state of the art hybrid DNN ASR system, system performance can be improved by 17% when augmented with the proposed e-vector.

Additionally, we also evaluate the usefulness of the proposed feature on blind noise condition classification, and obtained effective results.

The rest of this paper is organized as follows. Section 2 briefly introduces i-vectors. Next, the e-vector is proposed in Section 3, and two extraction methods are presented. We demonstrate e-vector adaptation experiments in Section 4, and e-vector environment classification experiments in Section 5. Finally, we conclude in Section 6.

2. E-vector extraction method

Inspired by i-vectors, we extract a feature in the total variability space, which we refer to as e-vectors, to capture the environmental noise variability in particular. We propose two ways for extracting e-vectors. One is through a dimensionality reduction via LDA, with appropriate environment labels. The other one is through a bottleneck NN, where the input targets are i-vectors, and output targets are the appropriate environment conditions.

2.1. LDA based

The i-vector representation we obtained so far is speaker and channel dependent. In order to compensate the within class inter-speaker variability and the session variability, we use Linear Discriminant Analysis (LDA) to find a low dimensional representation.

The idea behind this approach is to seek new orthogonal axes to better discriminate among different noise environments. The axes found must satisfy the requirement of maximizing noise condition variance but minimizing within class variance.

$$\boldsymbol{S}_b = \sum_{c=1}^{C} \left(\boldsymbol{i}_c - \overline{\boldsymbol{i}}\right)\left(\boldsymbol{i}_c - \overline{\boldsymbol{i}}\right)^t$$

$$\boldsymbol{S}_w = \sum_{c=1}^{C} \frac{1}{n_c} \sum_{n=1}^{n_c} \left(\boldsymbol{i}_n^c - \overline{\boldsymbol{i}}\right)\left(\boldsymbol{i}_n^c - \overline{\boldsymbol{i}}\right)^t$$

http://dx.doi.org/10.21437/Interspeech.2017-485

where $\bar{i}$ is the mean of i-vectors for each class, C is the number of classes, and n_c is the number of utterances for each class. In the case of i-vectors, the speaker population mean vector is equal to the null vector since, in factor analysis, these i-vectors have a standard normal distribution, which has a zero mean vector. The purpose of LDA is to maximize the Rayleigh coefficient. This maximization is used to define a projection matrix A composed by the best eigenvectors (those with highest eigenvalues) of the general eigenvalue equation:

$$\boldsymbol{S}_b\boldsymbol{v} = \boldsymbol{\Lambda}\boldsymbol{S}_w\boldsymbol{v}, \tag{1}$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues. The i-vectors are then submitted to the projection matrix obtained from LDA.

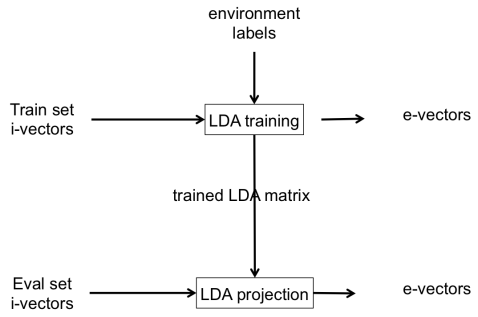

Figure 1: *Evector extraction via LDA using environment labels*

After the projection matrix is trained, it will be used for LDA projection on the eval set. Figure 1 depicts an overview of this process.

2.2. BN-NN based

A Bottleneck Neural-Network (BN-NN) refers to a particular topology of a NN, such that one of the hidden layers has significantly lower dimensionality than the surrounding layers. It is assumed that such a layer referred to as the bottleneck layer compresses the information needed for mapping the NN input to the NN output. A bottleneck feature vector is the vector of values at the bottleneck layer, as a by-product of forwarding a primary input feature vector through the BN-NN. In other words, after a BN-NN is trained for its primary task, the bottleneck layer is declared to be the output layer and all succeeding layers are ignored.

Following Sainath *et al.* we apply a low-rank approximation to the weights of the softmax layer of the network. This is done by replacing the usual softmax layer weights by a linear layer with a small number of hidden units, followed by a softmax layer. More specifically, a new BN output layer with r linear hidden units is inserted into the last weight matrix with a hidden layer of size h, and a softmax layer with s state posterior outputs. This changes the number of parameters from $h * s$ to $r * (h + s)$. There are two benefits of using this method. First, it ensures the best achievable frame accuracy even with a relatively small r. Second, the linearity of the output for the BN layer prevents any loss of information when we treat the DNN as a feature extractor.

The configuration for our BN-NN is 487x1024x1024xMxN, where M is the size of the bottleneck, and N is the number of targets. One of the key questions is what targets to train the BN features on. Since our goal is to obtain a bottleneck feature that is able to separate different noise conditions, the noise conditions are used as the targets.

Instead of a single bottleneck layer for all mixed noise conditions, our proposed BN-NN structure extracts separate BN-evectors for each individual noise condition, with the concatenated BN-Evector as the final bottleneck feature output. The extraction method is given in Figure 2. The bottleneck layers share the same weights for the preceding hidden layers, and each has its own softmax output with corresponding noise condition targets. Once we trained the BN-Evector extracting network using a certain noise environment label, we use the same network to generate BN-Evector for the eval set.

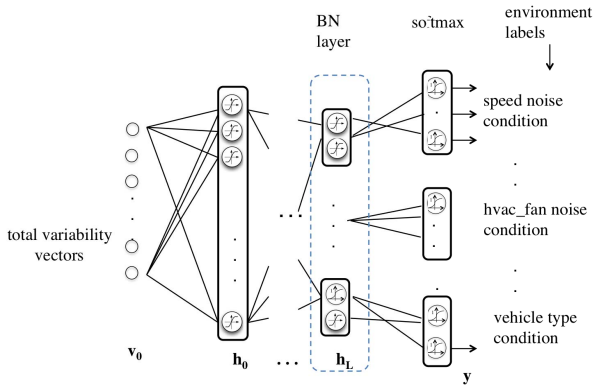

Figure 2: *Evector extraction via Bottleneck NN using environment labels*

3. E-vector for ASR adaptation

To evaluate the effectiveness of the proposed e-vector, we perform environment adaptation experiments by augmenting ASR system with e-vector. Similar to the noise adaptation in [9] using signal-to-noise ratio, and the speaker adaptation in [10] using i-vector, we append the e-vectors to original speech features to supply as an adaptive bias to the neurons in the first hidden layer.

3.1. Dataset

We perform the experiments on Ford Corpus [11]. Ford Corpus is a 30-hour, 5k-vocabulary dataset collected in a *real* driving, reverberant and noisy environment. The utterances were recorded on a real road, in vehicles of varying body styles (e.g., small, medium, large car, SUV, pick-up truck) with real speakers (drivers) with varying gender, age and dialects, under different ambient noise conditions (blower, road surface, vehicle speed, vehicle wipers on/off, vehicle windows open/closed, etc.). There are 3 speed conditions (0/35/65 MPH), 2 hvac_fan conditions (On/Off), 2 wiper conditions (On/Off), and 5 vehicle types in this dataset. For our experiments, the data were randomly partitioned into three sets with non-overlapping speakers. The training set contains 17,183 utterances from 90 speakers, the development set contains 2,773 utterances from 14 speakers, and the evaluation set contains 1,763 utterances from 9 speakers. Aside from the speakers, all other recording conditions are found in all three data sets.

3.2. Baseline model

Our baseline model follows the recipe in [11]. For training a GMM baseline system, we first flat-start trained 26 context-independent monophone acoustic models, then used these mod-

els to bootstrap the training of a context-dependent triphone GMM-HMM system. The resulting models contain 3,158 tied triphone states, and 90K Gaussians.

The hybrid DNN baseline is trained with 4 hidden layers, where each hidden layer has 1024 units; the DNN has 3158 output units. The input to the network consists of 11 stacked frames (5 frames on each side of the current frame). Both cross entropy (CE) and state-level minimum Bayes risk (sMBR) loss objectives are used. The stochastic gradient descent (SGD) used minibatches of 256 frames, and an exponentially decaying schedule that starts with an initial learning rate of 0.008 and halves the rate when the improvement in frame accuracy on a cross-validation set between two successive epochs falls below 0.5%. Cross-validation is done on a set of 180 utterances that are held out from the training data.

The speech features are fMLLR-adapted [12] MFCC-LDA-MLLT features, as in [11]. In our experiments fMLLR is applied both during training and test, which is known as speaker-adaptive training (SAT) [13].

The eval set baseline recognition results are presented in Table 1. For the rest of this paper, we choose the hybrid DNN-SMBR model as our baseline system, and all further results are based on this baseline.

Table 1: *ASR baseline WER. *For the rest of this paper, the DNN-sMBR system is used as our baseline model.*

Model	WER (%)
GMM	11.86
hybrid DNN-CE	7.04
hybrid DNN-sMBR*	6.53

3.3. LDA-Evector adaptation results

We can further improve the WER by augmenting original acoustic features with the e-vector we extracted. The system is adapted as follows. The speech features are derived from speech signals in the same way with baseline model. The DNN acoustic models input is a super vector with e-vector appended to the speech features.

The results of environment adaptation using lda-evector are displayed in Table 2. Our experiments show that all of the systems with e-vectors extracted from different noise targets improved the WERs. Here we only list the result of six good feature combination candidates in Table 2. We can see that the model trained with lda-evector extracted from [speed, hvac_fan, wiper, vehicle_type] improved the relative WER by 16% compared to the DNN-sMBR baseline, reducing WER from 6.53 to as low as 5.47.

Table 2: *ASR WER using LDA-Evectors*

LDA Projection Label used	Evector dim	WER
speed	2	6.31
hvac_fan	1	5.75
wiper	1	6.34
vehicle_type	4	5.91
speed+hvac_fan	5	5.73
speed+hvac_fan+wiper	11	5.48
speed+hvac_fan+wiper+vihecle_type	20	5.47

3.4. BN-Evector adaptation results

Table 3: *ASR WER using BN-Evectors*

BN-NN output target	BN-evector dim	WER
speed	10	6.28
hvac_fan	10	5.74
wiper	10	6.14
vehicle	10	5.81
speed+hvac_fan	20	5.67
speed+hvac_fan+wiper	30	5.47
speed+hvac_fan+wiper+vehicle_type	40	5.43

The results of environment adaptation using BN-Evector are displayed in Table 3. The BN-Evector is trained on an utterance base and is appended to each frame of the corresponding utterance. We can see that a 17% relative improvement is achieved, reducing WER from 6.53 to as low as 5.43. Comparing the results in Table 2 and Table 3, we observe that the BN-Evector performs slightly better than the lda-evector.

Table 4: *The effect of the size of BN Layer*

BN-NN output target	BN-evector dim	WER
speed	2	6.3
	5	6.28
	10	6.28
hvac_fan	2	5.76
	5	5.74
	10	5.74
wiper	2	6.17
	5	6.14
	10	6.14
vehicle	2	5.94
	5	5.82
	10	5.81

We also investigate the effect of the size of the bottleneck in the BN-NN, which directly influences the dimensionality of the resulting BN-Evector. Results in Table 4 show that the performance starts to saturate at around 10. In Table 3 and the rest of this paper, we keep our BN size to be 10 for each noise condition.

3.5. Comparison and Fusion

To better understand the performance, in Table 5 we evaluate the WER obtained by augmenting original acoustic features with raw i-vector. We can see that i-vector adaptation gives a 2.5% relative improvement. We also vary the i-vector dimensionality to evaluate its effect, and 92 gives the best performance, although this effect is very mininal.

Table 5: *ASR WER using ivectors directly*

Ivector dim	WER(%)
92	6.37
50	6.38

Table 6 compares the improvement brought by i-vector adaptation and e-vector adaptation. Both the lda-evector and BN-Evector outperform the i-vector by a large margin. This

indicates that e-vector is able to capture accurate information about the noise environment. The reason that i-vector is not as good might be the non-environment information might be redundant and could lead to a biased adaptation.

Table 6: *ASR WER comparison using ivector and proposed evector.*

System	WER(%)
baseline	6.53
+ivector	6.37
+lda-evector	5.47
+BN-Evector	5.43

Table 7 reports the ASR performance of fusing i-vector's and e-vector's for adaptation. The fusion result is slightly better than using the i-vector or e-vector individually. This indicates that i-vector and e-vector are complementary, although e-vector is trained from i-vector. In practice, the i-vector and e-vector can be used in conjunction with each other to achieve better adaptation results.

Table 7: *ASR WER fusing ivector and evector.*

System	WER(%)
baseline	6.53
+ivector+lda-evector	5.4
+ivector+BN-Evector	5.3

4. E-vector for noise environment identification

In this section, we explore the effectiveness of the e-vector on a different task, noise condition classification. We use the trained bn network to extract BN-Evector's from the eval audio files, and then use the BN-Evector to blindly classify certain noise condition of the corresponding audio. Table 8 reports the classification equal error rate (EER) on speed using BN-Evector. Table 9 reports the classification EER on hvac status using BN-Evector. Table 10 reports the classification EER on wiper status using BN-Evector. Table 11 reports the classification EER on vehicle type using BN-Evector. We can see that BN-evector is capable of classifying different noise conditions.

Table 8: *Speed classification EER using BN-Evector.*

BN-Evector extracted with labels	classification EER (%)
speed	19.5
speed+hvac_fan	19

Table 9: *Hvac status classification EER using BN-Evector.*

BN-Evector extracted with labels	classification EER (%)
hvac_fan	16.5
speed+hvac_fan	15.5

The results show that the e-vector is able to identify most of the noise conditions correctly. This indicates that the e-vector contain useful information about the noise environment. We also observe that, in all four experiments, classification accuracy is improved using e-vectors training with more noise labels.

Table 10: *Wiper status classification EER using BN-Evector.*

BN-Evector extracted with labels	classification EER (%)
wiper	18
wiper+speed_fan	17.5
wiper+speed+hvac_fan	16.5

Table 11: *Vehicle type classification accuracy using BN-Evector.*

BN-Evector extracted with labels	classification EER (%)
vehicle	21.5
vehicle+speed	21
vehicle+hvac_fan	20

5. Conclusion

We presented a feature representation trained from i-vector, which we refer to as e-vector, to capture the channel and environment variability specifically. We have given two e-vector extraction methods, one via LDA projection, and the other via a multi Bottleneck DNN. Our experiments on environment adaptation using the proposed e-vector brought a 17% WER improvement on real-recorded noisy corpora. This outperformed raw i-vector adaptation improvement by a large margin. We also showed that the result can be further improved when fusing i-vector's and e-vector's together. The extracted environment features can also be applied to the blind environment classification problem, and our experiment demonstrated the capability of separating noise conditions using the proposed e-vector. In the future, we could consider more noise labels, to train a more complete feature capturing the surrounding noise environment. This method can be applied to other noisy speech scenarios.

6. References

[1] C. Nikias and J. Mendel, "Signal processing with higher-order spectra," *Signal Processing Magazine*, vol. 10, no. 3, pp. 10–37, 1993.

[2] R. A. Wiggins, "Minimum entropy deconvolution," *Geoexploration*, vol. 16, no. 1, pp. 21–35, 1978.

[3] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification." in *Interspeech*, vol. 9, 2009, pp. 1559–1562.

[4] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[5] M. Senoussaoui, P. Kenny, N. Dehak, and P. Dumouchel, "An i-vector extractor suitable for speaker recognition with both microphone and telephone speech." in *Odyssey*, 2010, p. 6.

[6] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks." in *Interspeech*, vol. 237, 2011, p. 240.

[7] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[8] P. Cardinal, N. Dehak, Y. Zhang, and J. Glass, "Speaker adaptation using the i-vector technique for bottleneck features," in *Proceedings of Interspeech*, vol. 2015, 2015.

[9] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP*. IEEE, 2013, pp. 7398–7402.

[10] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors." in *ASRU*, 2013, pp. 55–59.

[11] X. Feng, B. Richardson, S. Amman, and J. Glass, "On using heterogeneous data for vehicle-based speech recognition: A dnn-based approach," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4385–4389.

[12] M. J. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[13] S. Matsoukas, R. Schwartz, H. Jin, and L. Nguyen, "Practical implementations of speaker-adaptive training," in *DARPA Speech Recognition Workshop*. Citeseer, 1997.